

SYLABUS ZAJ /GRUPY ZAJ

Dane ogólne:

Jednostka organizacyjna:	Katedra Informatyki				
Kierunek studiów:	Informatyka				
Specjalno /Specjalizacja:					
Nazwa zaj / grupy zaj :	Technologie obiektowe i komponentowe				
Course / group of courses:	Object and Component Technologies				
Forma studiów:	stacjonarne				
Nazwa katalogu:	WP-IN-I-20/21Z				
Nazwa bloku zaj :					
Kod zaj /grupy zaj :	106000	Kod Erasmus:			
Punkty ECTS:	4	Rodzaj zaj :		obowi zkowy	
Rok studiów:	2	Semestr:		4	
Rok	Semestr	Forma zaj	Liczba godzin	Forma zaliczenia	ECTS
2	4	LO	30	Zaliczenie z ocen	3
		W	15	Zaliczenie z ocen	1
Razem			45		4
Koordinator:	magister in ynier Rafał J dryka				
Prowadz cy zaj cia:					
J zyk wykładowy:	semestr: 4 - j zyk polski				

Obja nienia:

Rodzaj zaj : obowi zkowe, do wyboru.

Forma prowadzenia zaj : W - wykład, - wiczenia audytoryjne, L - lektorat, S – seminarium/ zaj cia seminaryjne, P - wiczenia praktyczne (w tym zaj cia wf), M - wiczenia specjalistyczne (medyczne/ kliniczne), LO – wiczenia laboratoryjne, LI - laboratorium informatyczne, ZTI - zaj cia z technologii informacyjnych, P – wiczenia projektowe, ZT – zaj cia terenowe, T - wiczenia terenowe na obozach programowych, SK - samokształcenie (i inne), PR - praktyka zawodowa

Dane merytoryczne

Wymagania wst pne:			
Znajomo algorytmów i struktur danych, umiej tno programowania proceduralnego oraz podstawowa znajomo technik obiektowych.			
Szczegółowe efekty uczenia si			
Lp.	Student, który zaliczył zaj cia zna i rozumie/potrafi/jest gotowy do:	Kod efektu dla kierunku studiów	Sposób weryfikacji efektu uczenia si
1	Ma wiedz dotycz c wykorzystywania zaawansowanych technik obiektowych i komponentowych w projektowaniu systemów informatycznych.	IN1_W07, IN1_W04	ocena aktywno ci
2	Ma wiedz w zakresie tworzenia oprogramowania dla ró nych zastosowa informatyki. Zna standardy i normy techniczne stosowane w informatyce, rozumie cykl ycia oprogramowania oraz etapy wytwarzania w zakresie projektowania, implementacji, testowania oraz wdó enia.	IN1_W08, IN1_W11	kolokwium
3	Potrafi korzysta z dokumentacji technicznej standardowych pakietów/bibliotek oraz komponentów reprezentuj cych struktury danych i algorytmy ich przetwarzania.	IN1_U01	kolokwium

4	Potrąfi zaprojektować hierarchie klas i pakietów w celu programowania komponentów wielokrotnego wykorzystania spełniających wymagania architektury trójwarstwowej i zadań przypisanych do każdej warstwy.	IN1_U05	kolokwium
5	Umie zaimplementować proste kody i komponenty oraz stworzyć własne pakiety/biblioteki, klasy i metody wchodzące w skład warstwy logiki aplikacji i warstwy dostępu do danych.	IN1_U05	kolokwium
6	Ma wiadomość o roli i znaczeniu wiedzy w społeczeństwie, gospodarce, firmach i organizacjach.	IN1_K02	kolokwium

Stosowane metody osiągnięcia zakładanych efektów uczenia się (metody dydaktyczne)

metody podaje (Wykład tradycyjny (informacyjny) z wykorzystaniem prezentacji (PP) i demonstracji przykładów.), metody praktyczne (ćwiczenia laboratoryjne prowadzone na podstawie przygotowanych instrukcji (udostępnianych w sposób elektroniczny na platformie e-learning) z przykładami i zadaniami do samodzielnej realizacji.)

Kryteria oceny i weryfikacji efektów uczenia się

wiedza:

- ocena kolokwium (ocena kolokwium)
- ocena aktywności (ocena aktywności na zajęciach)

umiejętności:

- ocena kolokwium (ocena kolokwium)

kompetencje społeczne:

- ocena kolokwium (ocena kolokwium)

Warunki zaliczenia

Kolokwia w formie praktycznych zadań programistycznych, kryteria oceny zgodne z obowiązującym Regulaminem studiów PWSZ w Tarnowie

Treści programowe (opis skrócony)

1. Zaawansowane techniki obiektowe.
2. Implikacje i problemy związane z mechanizmem dziedziczenia.
3. Komponenty - rodzaje, metody tworzenia i używania, definiowanie kontraktów.
4. Problemy związane z komponentami.
5. Kierunki rozwoju - programowanie aspektowe, wstrzykiwanie zależności.

Content of the study programme (short version)

1. Advanced object-oriented techniques.
2. Implications and problems related to the inheritance mechanism.
3. Components - types, methodologies of creating and using, defining contracts.
4. Component problems.
5. Development directions - aspect programming, dependency injection.

Treści programowe

	Liczba godzin
Semestr: 4	
Forma zajęć: wykład	
Podstawowe cele obiektowości, określenie odpowiedzialności obiektu, mechanizmy, które je wspierają (dziedziczenie, polimorfizm, hermetyzacja), omówienie ich zastosowań oraz porównanie możliwości, jakie oferują. Podstawowe kryteria jakości projektu obiektowego: spójność i powiązanie między obiektami. Wprowadzenie do programowania komponentowego. Koncepcja komponentu i kontenera komponentów, Omówienie sposobów wiązania komponentów ze sobą oraz osadzania ich w kontenerze komponentów. Techniki rozwiązywania zależności pomiędzy komponentami z wykorzystaniem wzorca Service Locator oraz poprzez ich wstrzykiwanie. Przykłady z wykorzystaniem SpringFramework. Infrastruktura Spring, mechanizm wstrzykiwania zależności, programowanie aspektowe, koncepcja fabryk komponentów. Zasady i założenia testowania jednostkowego wykorzystującego biblioteki JUnit Javy. Koncepcja obiektów zastępczych, które umożliwiają stosowanie testowania jednostkowego także w przypadku obiektów o złożonych zależnościach. Statyczne i dynamiczne techniki generowania obiektów zastępczych oraz sposoby ich wykorzystania.	15
Forma zajęć: wiczenia laboratoryjne	
Podstawowe cele obiektowości, określenie odpowiedzialności obiektu, mechanizmy, które je wspierają (dziedziczenie, polimorfizm, hermetyzacja), omówienie ich zastosowań oraz porównanie możliwości, jakie oferują. Podstawowe kryteria jakości projektu obiektowego: spójność i powiązanie między obiektami. Wprowadzenie do programowania komponentowego. Koncepcja komponentu i kontenera komponentów,	30

Omówienie sposobów wiązania komponentów ze sobą oraz osadzania ich w kontenerze komponentów. Techniki rozwiązania zależnościami komponentami z wykorzystaniem wzorca Service Locator oraz poprzez ich wstrzykiwanie. Przykłady z wykorzystaniem SpringFramework. Infrastruktura Spring, mechanizm wstrzykiwania zależnościami, programowanie aspektowe, koncepcja fabryk komponentów. Zasady i założenia testowania jednostkowego wykorzystujących bibliotek JUnit Javy. Koncepcja obiektów zastępczych, które umożliwiają stosowanie testowania jednostkowego także w przypadku obiektów o złożonych zależnościach. Statyczne i dynamiczne techniki generowania obiektów zastępczych oraz sposoby ich wykorzystania.	30
---	----

Literatura
Podstawowa
B. McLaughlin, G. Pollice, D. West, Head First Object-Oriented Analysis and Design (edycja polska)
W. Wheeler, J. White, Spring w praktyce
Tutoriale dostępne na oficjalnej stronie SpringFramework
Uzupełniająca

Dane jakościowe

Przyporządkowanie zajęć /grup zajęć do dyscypliny naukowej/artystycznej		informatyka techniczna i telekomunikacja	
Sposób określenia liczby punktów ECTS			
Forma nakładu pracy studenta (udział w zajęciach, aktywność, przygotowanie sprawozdania, itp.)		Obciążenie studenta [w godz.]	
Udział w zajęciach		45	
Konsultacje z prowadzącym		2	
Udział w egzaminie		0	
Bezpośredni kontakt z nauczycielem - inne		0	
Przygotowanie do laboratorium, wicze, zajęcia		20	
Przygotowanie do kolokwium i egzaminu		13	
Indywidualna praca własna studenta z literatury, wykładami itp.		20	
Inne		0	
Sumaryczne obciążenie prac studenta		100	
Liczba punktów ECTS			
Liczba punktów ECTS		4	
Zajęcia wymagające bezpośredniego udziału nauczyciela akademickiego		L. godzin	ECTS
		47	1,9
Zajęcia o charakterze praktycznym		L. godzin	ECTS
		60	2,4

1 godz = 45 minut; 1 punkt ECTS = 25-30 godzin

W sekcji 'Liczba punktów ECTS' suma punktów ECTS zajęć wymagających bezpośredniego udziału nauczyciela akademickiego i o charakterze praktycznym może się różnić od łącznej liczby punktów ECTS dla zajęć/grup zajęć.